

# Spinner

Spinner A Spinner is a loading indicator that provides visual feedback to users during asynchronous operations. It helps maintain user engagement by clearly communicating that a process is in progress, reducing perceived wait times and improving user experience. Import ■ `import { AavaSpinnerComponent } from "@aava/play-core"` ; Basic Usage ■ The spinner component supports multiple visual styles, sizes, and colors to match your application's design system. Sizes ■ Available spinner sizes: xs (Extra Small) : Smallest spinner for inline use sm (Small) : Compact spinner for inline use md (Medium) : Default size for general use lg (Large) : Prominent spinner for important operations xl (Extra Large) : Maximum size for high-impact loading states Colors ■ Semantic color variants: primary : Default brand color secondary : Secondary brand color success : Success state indication warning : Warning state indication danger : Error or critical state indication Accessibility ■ Built-in accessibility features ensuring inclusive user experience for loading states. Accessibility Features ■ ARIA Labels : Use `aria-label` or `aria-labelledby` to describe the loading state Live Regions : Announce loading state changes to screen readers using `aria-live` Focus Management : Ensure proper focus handling during loading states Reduced Motion : Respect user preferences for reduced motion with `prefers-reduced-motion` Timeout Handling : Provide fallback mechanisms for extended loading times Screen Reader Support : Semantic HTML structure for assistive technologies Keyboard Navigation : Maintain keyboard accessibility during loading states Color Contrast : Ensure sufficient contrast for all spinner variants Status Communication : Clear communication of loading progress and completion API Reference ■ Inputs ■ Property Type Default Description type `SpinnerType` 'circular' Visual style of the spinner size `SpinnerSize` 'md' Size of the spinner color `SpinnerColor` 'primary' Color variant of the spinner animation boolean true Whether to animate the spinner progressIndex number undefined Progress value for determinate loading (0-100) CSS Custom Properties ■ Size Tokens ■ Property Description Default `--spinner-size-xs` Extra small spinner dimensions 16px `--spinner-size-sm` Small spinner dimensions 20px `--spinner-size-md` Medium spinner dimensions 24px `--spinner-size-lg` Large spinner dimensions 48px `--spinner-size-xl` Extra large spinner dimensions 64px Color Tokens ■ Property Description Default `--spinner-primary-track` Primary spinner track color `rgba(59, 130, 246, 0.2)` `--spinner-primary-fill` Primary spinner fill color `rgb(59, 130, 246)` `--spinner-secondary-track` Secondary spinner track color `rgba(107, 114, 128, 0.2)` `--spinner-secondary-fill` Secondary spinner fill color `rgb(107, 114, 128)` `--spinner-success-track` Success spinner track color `rgba(34, 197, 94, 0.2)` `--spinner-success-fill` Success spinner fill color `rgb(34, 197, 94)` `--spinner-warning-track` Warning spinner track color `rgba(245, 158, 11, 0.2)` `--spinner-warning-fill` Warning spinner fill color `rgb(245, 158, 11)` `--spinner-error-track` Error spinner track color `rgba(239, 68, 68, 0.2)` `--spinner-error-fill` Error spinner fill color `rgb(239, 68, 68)` Animation Tokens ■ Property Description Default `--spinner-animation-duration` Spinner rotation duration 3s `--spinner-animation-timing` Spinner animation timing function linear Best Practices ■ Design Guidelines ■ Context

Appropriate : Use appropriate sizes for the context and available space Semantic Colors : Choose colors that align with your design system and semantic meaning Prominent Placement : Position spinners prominently for critical operations Loading States : Consider using skeleton screens for complex loading states Clear Context : Provide clear context about what is loading Consistent Timing : Use consistent animation timing across your application Visual Hierarchy : Size spinners according to the importance of the loading operation Performance ■ Timing Thresholds : Only show spinners for operations that take more than 200ms Animation Optimization : Use CSS animations instead of JavaScript for better performance Progress Indicators : Use progress mode for operations with known duration State Management : Implement proper loading state management to prevent flickering Resource Efficiency : Avoid unnecessary re-renders during loading states Bundle Size : Consider lazy loading spinner variants not immediately needed User Experience ■ Clear Messaging : Provide clear messaging about what is loading Consistent Patterns : Use consistent spinner styles throughout your application Cancel Options : Consider providing cancel options for long-running operations Error Handling : Implement proper error handling for failed operations Progress Feedback : For long operations, show progress or estimated time Completion States : Provide clear indication when loading is complete

```
<aava-spinner  
  type="circular"  
  color="primary"  
  size="lg"  
  [animation]="true"  
></aava-spinner>
```

```
<aava-spinner
  type="circular"
  color="primary"
  size="xs"
  [animation]="true"
></aava-spinner>
```

```
<aava-spinner
  type="circular"
  color="primary"
  size="sm"
  [animation]="true"
></aava-spinner>
```

```
<aava-spinner
  type="circular"
  color="primary"
  size="md"
  [animation]="true"
></aava-spinner>
```

```
<aava-spinner
  type="circular"
  color="primary"
  size="lg"
  [animation]="true"
></aava-spinner>
```

```
<aava-spinner
  type="circular"
  color="primary"
  size="xl"
  [animation]="true"
></aava-spinner>
```

■ No code found